Performance Testing Architecture for Communication Protocols

János Zoltán Szabó

In today's telecommunication protocols and applications it is important to check not only the functional correctness of implementations, but their performance characteristics as well. The purpose of performance testing is to verify whether the tested system can work under realistic load conditions and handle overloaded situations. The usual technique for testing the performance of end-to-end services and applications models each service user with a probabilistic timed state machine ([1]). During the test execution a large number of these simple automata are run in parallel by the test environment.

A new test language, the third edition of Tree and Tabular Combined Notation (TTCN-3, [2]), which was recently standardized by ETSI, can be efficiently used to specify such performance test scripts. The built-in TTCN-3 language constructs makes it possible to create dynamically any number of parallel test executor processes, the so called Parallel Test Components (PTCs). Performance testing requires at least as much computational resources for the tester as the tested implementation. Therefore the testing of distributed implementations is feasible only with distributed test environments.

This paper presents our basic ideas for a TTCN-3 based parallel and distributed performance testing prototype environment. The testing system consists of several parallel processes and a control protocol. The behaviour of each PTC is realized in a separate process. Each computer that takes part in test execution runs a special process called Host Controller (HC), which is responsible for the creation of new PTCs on that host. In addition, there is a dedicated process (a so called Main Controller, MC), which provides user interface and performs the tasks that require central coordination, such as load balancing among the computers.

The control protocol between the processes uses reliable transport layer connections (e.g. TCP connections). It covers all parallelism related TTCN-3 operations, such as PTC creation and termination or establishment of internal communication channels between PTCs. The protocol is platform independent, so a group of computers with heterogeneous hardware and operating systems can cooperate and generate load simultaneously.

The key aspects of design were the scalability, the robustness and the execution speed. The test architecture has been successfully implemented as an extension for an existing, compiler based test executor, which translates TTCN-3 test specifications into C++ programs. We have successfully applied the test environment for performance evaluation of two IP mobility protocols.

References

[1] M. Kwiatkowska, G. Norman, R. Segala, J. Sproston: Verifying Quantitative Properties of Continuous Probabilistic Timed Automata, CONCUR'2000, 2000.

[2] Methods for Testing and Specification (MTS); The Tree and Tabular Combined Notation version 3. TTCN-3: Core Language. ETSI ES 201 837-1.